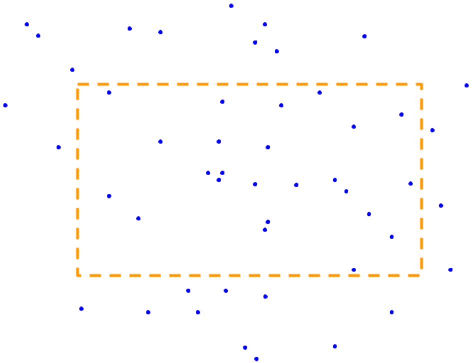
Depth Estimation via Sampling

Nikhilesh Rajaraman

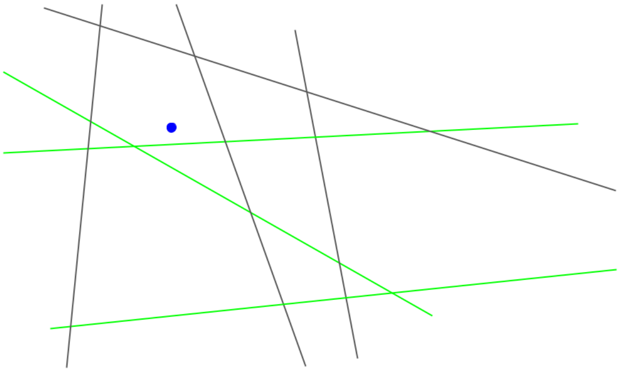
November 2024

Range Counting



- **Goal:** For any query set (or range), count the number of points inside.
- We are allowed to **pre-process** the input set of points to make counting queries efficient.
- Sets can be halfplanes, rectangles, disks, etc. or their higher dimensional analogues.

Planar Point Location



- **Goal:** For any query point, count the number of lines below it.
- Again, we can **pre-process** the input set of lines to make counting queries efficient.

Depth Calculation

- Generalizes the range counting and point location problems.
- $\mathcal{S}$: Finite set of **objects** with $|\mathcal{S}| = n$.
- $\mathcal{R}$: Set of allowed **ranges.** (may be infinite)
- The **depth** of a range $R \in \mathcal{R}$ - denoted μ_R - is the number of objects in $\mathcal{S}$ that intersect R.
- **Goal:** Given a set of n objects, pre-process them into a data structure to efficiently answer depth queries for ranges in $\mathcal{R}$.

Hardness of Depth Calculation

- There is an inherent trade-off between **query time** and **space** occupied by the data structure for answering depth queries.
- When $\mathcal{R}$ consists of halfspaces in $\mathbb{R}^d$ and objects are points, the worst case query time using m space data structures is lower bounded by $O(n/m^{1/(d+1)})$. (Arya, Mount, 2012)
- In $\mathbb{R}^2$, we therefore require at least n^3 space to achieve polylogarithmic query time for halfplanes, and can only guarantee $O(n^{2/3})$ query time using linear space.

Depth Estimation

- We can get around the lower bound by looking at the approximate version of the problem instead.
- **Input:** Set of n objects $\mathcal{S}$, set of ranges $\mathcal{R}$, parameter $\varepsilon \in [0, 1]$.
- **Goal:** Given a query range $R \in \mathcal{R}$, find a number α_R such that $(1 - \varepsilon)\mu_R \leq \alpha_R \leq \mu_R$.
- For a fixed ε, the goal is to pre-process the set of objects into a data structure to efficiently answer depth estimation queries.
- **Holy Grail:** For any fixed ε, polylogarithmic query time with linear space for halfspace range counting queries.

Decision Version of Depth Estimation

- As a first step, we will look at the decision version of the depth estimation problem.
- **Input:** Set of n objects $\mathcal{S}$, set of ranges $\mathcal{R}$, parameter $\varepsilon \in [0, 1]$, threshold $z \in [0, n]$.
- **Goal:** Given a query range R, output the following:
  1. If $\mu_R \geq (1 + \varepsilon)z$, output 1 .
  2. If $\mu_R \leq (1 - \varepsilon)z$, output 0.
  3. If $\mu_R \in [(1 - \varepsilon)z, (1 + \varepsilon)z]$, output either 0 or 1.

Depth Estimation Reduces to the Decision Version

- **Claim:** Depth estimation reduces to $O\left(\log\left(\varepsilon^{-1}\log(n)\right)\right)$ calls of the decision version.
- **Idea:** Use binary search over an exponentially spaced grid of thresholds.
- Let $z_i = (1+\varepsilon)^i$ for $i \leq \log_{1+\epsilon}(n) = O(\varepsilon^{-1}\log(n))$ be the grid of thresholds.
- Let i^* be such that the decision algorithm outputs 0 with threshold z_{i^*} and outputs 1 with threshold z_{i^*-1}.
- Easy to see that the true depth lies in $[(1-\varepsilon)z_{i^*}, z_{i^*}]$, and so z_{i^*} is a valid estimate for the depth with error within an ε factor.
- i^* can be found using at most $O\left(\log\left(\varepsilon^{-1}\log(n)\right)\right)$ calls to the decision algorithm using binary search.

Range Emptiness

- We will now consider an even simpler problem - that of determining whether a range is empty or not.
- **Input:** Set of n objects $\mathcal{S}$, set of ranges $\mathcal{R}$.
- **Goal:** Given a query range $R \in \mathcal{R}$, determine whether $\mu_R > 0$ or whether $\mu_R = 0$
- Range emptiness is a well studied problem - for halfspace ranges in $\mathbb{R}^d$, there exists a linear space data structure that can answer emptiness queries in $O(\log(n))$ time. (Dobkins, Kirkpatrick, 1985)
- We will show that solving depth estimation actually reduces to the emptiness problem!

From Range Emptiness to Depth Estimation

- **Goal:** Given a query range R and a threshold z, correctly determine whether $\mu_R \geq (1+\varepsilon)z$ or $\mu_R \leq (1-\varepsilon)z$.
- **Idea:** If a range has low depth, the probability that it contains no points from a random subsample of $\mathcal{S}$ is high, and vice versa.
- Let $B \subseteq \mathcal{S}$ be a random sample obtained by sampling each object in $\mathcal{S}$ independently with probability $p = 1/z$.
- Let p_{empty} be the probability that R intersects with no objects in B.

$$p_{\text{empty}} = \left(1 - \frac{1}{z}\right)^{\mu_R}$$

From Range Emptiness to Depth Estimation

$$p_{\text{empty}} = \left(1 - \frac{1}{z}\right)^{\mu_R}$$

- For a fixed threshold z, p_{empty} is a function of only the depth μ_R. Thus, we can **estimate** μ_R by estimating p_{empty} instead.
- **Estimating p_{empty}**:
  1. Construct i.i.d. random samples $B_1, B_2, \ldots, B_M$ for sufficiently large M.
  2. Run emptiness queries for range R on each sample.
  3. Calculate the fraction of queries that are empty.

Depth Estimation Algorithm (Aronov, Har-Peled, 2008)

Inputs: Set of objects $\mathcal{S}$, threshold z, error tolerance ε.

Data Structure:

- Set $M = c\varepsilon^{-2}\log(n)$
- Generate M i.i.d. samples $B_1, B_2, \ldots, B_M$. B_i is generated by picking each object in $\mathcal{S}$ with probability $1/z$.
- Construct M emptiness query data structures for the samples $B_1, B_2, \ldots, B_M$.

Answering a query for a range R:

- Run emptiness queries for R on each of the M samples.
- Calculate $\hat{p}_{\text{empty}} = n_{\text{empty}}/M$, where n_{empty} is the number of samples in which R was empty.
- Check if $\hat{p}_{\text{empty}} \leq (1 - 1/z)^z$. If yes, output $\mu_R \geq z$. Else, output $\mu_R < z$.

Correctness

Lemma: With high probability, the algorithm is correct whenever $\mu_R \notin [(1-\varepsilon)z, (1+\varepsilon)z]$.

Proof:

- We will consider the case $\mu_R < (1-\varepsilon)z$ and show that the algorithm outputs correctly in this case - the other case $\mu_R > (1+\varepsilon)z$ is similar.
- If $\mu_R < (1-\varepsilon)z$, we have $p_{\text{empty}} > (1-1/z)^{(1-\varepsilon)z}$
- Only makes a mistake if $\hat{p}_{\text{empty}} \leq (1-1/z)^z \leq (1+\Omega(\varepsilon))p_{\text{empty}}$
- Chernoff Bound:

$$\mathbb{P}\left\{\hat{p}_{\text{empty}} \geq (1+c\varepsilon)p_{\text{empty}}\right\} \leq \exp\left(-c' M\varepsilon^2 p_{\text{empty}}\right)$$

- Since p_{empty} is at least a constant, setting $M = \Omega(\varepsilon^{-2}\log(n))$ makes the error probability $1/\text{poly}(n)$, so the algorithm succeeds w.h.p.

Space and Time Analysis for Halfplanes

- **Recall:** Exact depth calculation for halfplanes in polylog time requires cubic space.
- Emptiness queries for halfplanes can be done in $O(\log(n))$ time using $O(n)$ space. (Dobkins, Kirkpatrick, 1985)
- For the decision version of depth estimation, the algorithm requires:
  - **Space:** $O(\varepsilon^{-2} n \log(n))$
  - **Query Time:** $O(\varepsilon^{-2} \log^2(n))$
- For the estimation problem:
  - **Space:** $O(\varepsilon^{-3} n \log^2(n))$
  - **Query Time:** $O(\varepsilon^{-2} \log^2(n) \log(\varepsilon^{-1} \log(n)))$
- Overall, the algorithm achieves **near-linear** space and **polylogarithmic** query time for fixed ε.

References and Resources

- *Geometric Approximation Algorithms* - S. Har-Peled
- *Tight Lower Bounds for Halfspace Range Searching* - S. Arya, D.M. Mount, 2012
- *A linear algorithm for determining the separation of convex polyhedra* - D.P. Dobkin, D.G. Kirkpatrick, 1985
- *On approximating the depth and related problems* - B.Aronov, S. Har-Peled, 2008